

8月18日

## 学习笔记

e2e: 端到端: 端到端是网络连接。网络要通信, 必须建立连接, 不管有多远, 中间有多少机器, 都必须在两头 (源和目的) 间建立连接, 一旦连接建立起来, 就说已经是端到端连接了, 即端到端是逻辑链路, 这条路可能经过了很复杂的物理路线, 但两端主机不管, 只认为是有两端的连接, 而且一旦通信完成, 这个连接就释放了, 物理线路可能又被别的应用用来建立连接了。TCP就是用来建立这种端到端连接的一个具体协议, SPX也是。

## hadoop、hbase、zookeeper关系

**hadoop**是**分布式**系统的一个文件系统, 主要有3部分分别是hdfs, yarn, mapreduce

hdfs hdfs是hadoop管理储存的实现。hdfs概念 数据块: 就像单机系统 (一台PC) 上的数据块一样, 提供磁盘读写的最小单位, 也就是磁盘读某个数据时会把改块的数据一次性全部读入。不过单机系统上的数据块大小为512字节, hdfs数据块大小为128MB。namenode和datanode: 分布式系统储存和单机储存另一个不同就是, 读取/写入某个数据时, 分布式系统需要确定这个数据放在哪个主机。这也是namenode和datanode解决的问题。

namenode储存着这些文件的具体位置 (具体在哪个主机和哪个块) 和元数据 (文件的一些属性), datanode是hdfs的工作结点, 负责储存并检索数据块。因为namenode如果损坏后整个系统就不能使用, 所以为了高可用性secondnamenode出现了, 它相当于namenode的副本。

**Hbase**是一个面向列族的存储器, 即Hbase在存储的时候将所有的列族成员都一起放在**HDFS**中存储, 存储图片的数据比较大, 图片的元数据比较少, 所以分成两个列族来储存。

**zookeeper**的一个应用场景: 有一组服务器提供某种服务, 我们希望客户端都能找到其中一台服务器, 然后我们需要维护这组服务器的成员列表, 这个列表不能在某个服务器上, 来避免单点故障, 并且如果某个服务器出现故障, 那么就需要从列表中删除改节点。这个场景不是一个被动的分布式结构, 它能够在某个外部事件发生主动的修改数据结构, zookeeper提供的就是这种服务。下面介绍它是如何实现这种应用的。

flink官网: [https://nightlies.apache.org/flink/flink-docs-release-1.10/zh/flinkDev/ide\\_setup.html](https://nightlies.apache.org/flink/flink-docs-release-1.10/zh/flinkDev/ide_setup.html)

flink github学习项目: <https://github.com/zhisheng17/flink-learning>

## spark和flink的区别

### 1) 设计理念

1、Spark的技术理念是使用微批来模拟流的计算, 基于Micro-batch, 数据流以时间为单位被切分为一个个批次, 通过分布式数据集RDD进行批量处理, 是一种伪实时。

2、Flink是基于事件驱动的, 是面向流的处理框架, Flink基于每个事件一行一行地流式处理, 是真正的流式计算. 另外他也可以基于流来模拟批进行计算实现批处理。

### (2) 架构方面

1、Spark在运行时的主要角色包括: Master、Worker、Driver、Executor。

2、Flink 在运行时主要包含: Jobmanager、Taskmanager和Slot。

### (3) 任务调度

1、Spark Streaming 连续不断的生成微小的数据批次，构建有向无环图DAG，根据DAG中的action操作形成job，每个job有根据窄宽依赖生成多个stage。

2、Flink 根据用户提交的代码生成 StreamGraph，经过优化生成 JobGraph，然后提交给 JobManager进行处理，JobManager 会根据 JobGraph 生成 ExecutionGraph，ExecutionGraph 是 Flink 调度最核心的数据结构，JobManager 根据 ExecutionGraph 对 Job 进行调度。

#### (4) 时间机制

1、Spark Streaming 支持的时间机制有限，只支持处理时间。使用processing time模拟event time必然会有误差，如果产生数据堆积的话，误差则更明显。

2、flink支持三种时间机制：事件时间，注入时间，处理时间，同时支持 watermark 机制处理迟到的数据,说明Flink在处理乱序大实时数据的时候,更有优势。

#### (5) 容错机制

1、SparkStreaming的容错机制是基于RDD的容错机制，会将经常用的RDD或者对宽依赖加Checkpoint。利用SparkStreaming的direct方式与Kafka可以保证数据输入源的，处理过程，输出过程符合exactly once。

2、Flink 则使用两阶段提交协议来保证exactly once。

#### (6) 吞吐量与延迟

1、spark是基于微批的,而且流水线优化做的很好,所以说他的吞入量是最大的,但是付出了延迟的代价,它的延迟是秒级;

2、而Flink是基于事件的,消息逐条处理,而且他的容错机制很轻量级,所以他能在兼顾高吞吐量的同时又有很低的延迟,它的延迟能够达到毫秒级;

## 心得

### 重大故障管理规程

目的：明确重大故障定义、规范重大故障处理流程，知道故障发生部门及时进行故障的分析与定位给、及时回复生产、并对故障进程复盘、排查、分析，降低故障发生频率，追求生产系统零故障，提升生产效率。

《重大故障管理规程》是为了明确重大故障定义、规范重大故障处理流程，知道故障发生部门及时进行故障的分析与定位给、及时回复生产、并对故障进程复盘、排查、分析，降低故障发生频率，追求生产系统零故障，提升生产效率。这门课程中也提到了项目成员、项目负责人等角色在发生故障之后，需要做的事情。故障紧急，是一个紧急预案，学习这门课程可以在关键时刻给予一个标准化的处理流程。因为在紧急时刻，人都是慌慌张张，脑子一热，大脑一片空白，根本不知道应该接下来做什么。在发生故障超过一定时间没有进行处理的，做出动作的，将纳入考核，这是对我们的一种激励，以免出现不作为的现象。这门课的学习，告诉了我公司系统出现重大故障的时候应该怎么做，做什么，怎么样做才能做到最好。这门课程中，关于出现重大问题的各种措施，我一定牢记心中，尽我最大的努力做到零故障，即使出现了故障，也要尽最大的努力，快速做出反应，给公司，给客户的损失降到最少，甚至没有。

### 生产运营违规（13类问题）讲解

《生产运营违规（13类问题）》这门课讲解了在我们日常的上班过程中，经常出现的问题。作为一名思特奇人，应当诚实守信，忠于职守。在这13类问题中，处罚的条目都是涉及到诚实守信、忠于职守的问题。比如请假领不住、法定节假日领补助、外出申请带考勤。带考勤肯定是员工不诚实守信，虚假报工，领补助和法定节假日领补助也一样，没有工作，却欺骗公司自己工作了，需要发放补助。又比如报工重复，报工不足等问题，自己干的工作不上报，或者说改做的工作没有完成才会出现这类问题，那么这肯定是不忠于职守的表现。这门课程，规范了我的行

为，用实际的例子，和应有的处罚规范我，提醒我在日常工作中的行为规范，教会了我如果做一名非常优秀、非常规范的思特奇人。

## **生产操作规程**

经过在公司近一个月的学习，我了解到了生产操作规范的重要性。在这门课程中，我学到了生产环境的操作流程、动作的规范性。这让我了解到了在生产环境中应当注意什么，做什么。比如常规在linux认为正常的不能再正常的操作，在生产环境的时候要格外的注意，不能用kill、mv、vi等。在这门课程中，我还学到了不同生产账号的作用，了解到了项目经理，运营组织等角色在生产环境中的职责，扮演的角色。这能使我在今后的工作中，遇到自己没有权限，解决不了的问题的时候，可以找对应的负责人解决问题。之后，又学到了生产账号的申请，课程中强调了生产账号管理和使用原则，虽然我现在还没有生产账号，但正是因为提前学习，了解使用方法，才能在将来的工作中，不出错不掉链子，全心工作。学习此门课程，最大的感悟就是，在生产环境中，要小心再小心，遇到问题立马上报，切不可自己做觉得，以免给公司造成不可挽回的损失。